//THIS IS FOR THAT HOMEWORK PROJECT 3

//function you can use

while(!s.emptyStack())

{

int x = s.popStack();

cout<<x<<” “;

}

to select randomize letters use the asci character number

Letter\_\_\_\_\_\_\_\_\_\_\_ASCII

‘A’ 65=random version is 65+0

‘B’ 66=65+1

‘C’

.

.

.

‘Z’ =65+25

^ 🡪 ^ 🡪 cout<<int(‘A’)=65

cout<<char(65)=A

x=rand()%26 🡨 0-25 randomize

ex. alphabets.pushStack(char(x+65));

//for months to randomize em

string months[12]={“Jan”,”Feb”,…”Dec”};

month[0] month[1] month[2]. . . . month[11]

Jan Feb Mar Dec

x.rand()%12 🡪 from 0 to 11

month.pushStack(months[x]);

if(x<=9) cout<<x;

else switch(x)

{

case 10: cout<<’A’; break;

case 11: cout<<’B’; break;

.

.

.

case 15: cout<<’F’l break;

}

if(x<=9) cout<<x;

else cout<<char(x+55);

STACK UPP, LOW, VOW; //uppercase, lowercase, vowel

class STACK

{

private: char a[80];

int counter;

}

//at the beginning we have to clear the stack

UPP.clearStack();

LOW.clearStack();

VOW.clearStack();

bool isVowel(char x)

{

x=toUpper(x);

if(x==’A’||x==’E’||x==’I’||. . .x==’U’)

return true;

else

return false;

} //this function introduced before the class

char c;

cout<<”enter a sentence. . .”<<endl;

while(cin.get(c), c!=’\n’)

{

if(isupper(c))UPP.pushStack(c);

if(islower(c)) ~ ~ ~

if(isVowel(c))~ ~ ~

}

class STACK

{

private:

int a[5];

int counter;

public:

void clearStack()

{counter = 0;}

bool emptyStack()

}

//note example of terniary

P: Operators

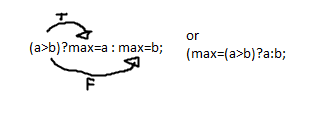
if(a>b)

max=a;

else

max=b;

//above is same as (a>b)?



//going back to above class stack

class STACK

{

private:

int a[5];

int counter;

public:

void clearStack()

{counter = 0;}

bool emptyStack()

{return(counter==0)?true:false;)}

bool fullStack()

{return (counter==5)?true:false;}

void pushStack(int x)

{a[counter]=x; counter++;}

int popStack()

{//the counter is 1 more than the index of the top element

counter--; return a[counter];

}//end popStack

}//end class STACK

int main()

{

STACK s; //in object s we have array of size 5 and a counter

}

//\*\*\*\*\*\*\*now we change it to a template function ayo don’t use this for the project due Wednesday sept 19

template <class T, int n>

class STACK 🡨 dunno if this disappears tho

{

private:

T a[n]; //now type template

int counter;

public:

void clearStack()

{counter = 0;}

bool emptyStack()

{return(counter==0)?true:false;)}

bool fullStack()

{return (counter==n)?true:false;}

void pushStack(T x)

{a[counter]=x; counter++;}

T popStack()

{//the counter is 1 more than the index of the top element

counter--; return a[counter];

}//end popStack

}//end class STACK

//the changes we have to make in main

int main()

{

//STACK < > p; //object p we want it for example char a[10]

//in the <> we put in the type and then the size

STACK <char, 10> p;

//example in object q we want array of size and string type string a[5]

STACK <string, 5> q;

}

/\*we can actually make a userdefined library. if we like it we can just keep calling this. . .

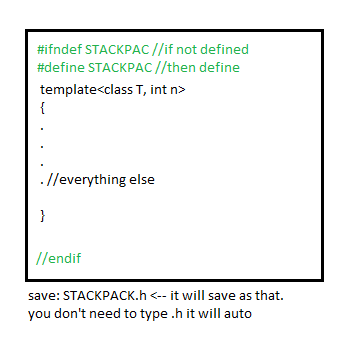
\*/

copy that entire template<class T, int n> stack thing

.cpp

.h

in visual studio add it in the header area add a header extension file of .h



now in our program we’ll need to include one line

example:

#include <iostream>

//you want to include your own library

#include “STACKPACK.h”

int main()

{

STACK <char, 10> p;

STACK <string, 5> q;

}

**//AYO DO THAT BEFORE WEDNESDAY SEPT 19 TO GET DAT EXTRA CREDIT BOI**

btw all of that above with the .h, that is called a userdefined library…

//going back to the stack in main

int main()

{

STACK <char, 10> p;

STACK <string, 5> q;

p.clearStack();

q.clearStack(); //we want to clear them first

for(int i=1; i<=10; ++i)

{

p.pushStack(char(i+64) p) //i+64 bec char 65 is ‘A’

}

}

**//AYO DO THAT BEFORE WEDNESDAY SEPT 19 TO GET DAT EXTRA CREDIT BOI**

**//display days of the week Monday to Friday n stuff.**

Back to Big-OH :’(

we use big oh to estimate runtime in mathematics

how many times a loop is going to execute

Recall

i.) for(int i=1; i<=n; ++i)

{

} 🡨 this is all O(n)

ii.) for(int i=1; i<=n; ++i) 🡪 O(n)

cout<<I;

for(int j=1; j<=n; ++j) 🡪 O(n)

cout<<j;

* O(n)+O(m) = O(max(n,m)) //whichever is bigger

iii.) for(int i=1; i<=n; ++i) 🡪 O(n)

{

for(int j=1; j<=n; ++j) 🡪 O(n)

{

for(int j=1; j<=n; ++j) 🡪 O(n)

cout<<i+j+k;

}

}

=O(n)\*O(n)\*O(n) = O(n^3)

//when you nest the loops, the program will run or take longer

//NOTE

f(n) = log (n^k+. . .)

f(n) = O(log n) //if log, doesn’t matter it is log of n I guess. . .

Find the big-OH estimate of the following

1. f(n) = (n+1)^4 (n-1)^3 + (n+2)^6

O(n^4) \* O(n^3) + O(n^6)

O(n^7) + O(n^6)

O(max(n^7, n^5)

=O(n^7)

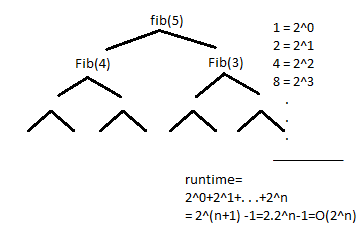
1. long int Fib(int n)

{

if(n<=2) return 1;

else return Fib(n-1)+Fib(n-2);

} //recursive function to return the nth Fibonacci number



//this is recursive Fibonacci, we will switch it to non recursive

1. NON RECURSIVE function to return the nth Fibonacci number

long int NRFib(int n)

{

//fib numbers, 1 1 2 3 5 8 we call 1, to be f0, 2nd 1 to be //f1, 2 to be fn=f0+f1, the old f1 becomes f0, the old fn //becomes f2=fn

long int f0=1, f1=1, fn;

for(inti =2; i<=n; ++i)

{

fn=f0+f1;

f0=f1;

f1=fn;

}

return fn; //this entire function big oh is O(n)

}

//**DUE WEDNESDAY September 19! LET’S MAKE BOTH RECURSIVE AND NON RECURRSIVE HOMIE!!**